PSYCHOMETRIKA—VOL. 62, NO. 2, 251–266
JUNE 1997

WEIGHTED LEAST SQUARES FITTING USING ORDINARY LEAST SQUARES ALGORITHMS

HENK A. L. KIERS

UNIVERSITY OF GRONINGEN

A general approach for fitting a model to a data matrix by weighted least squares (WLS) is studied. This approach consists of iteratively performing (steps of) existing algorithms for ordinary least squares (OLS) fitting of the same model. The approach is based on minimizing a function that majorizes the WLS loss function. The generality of the approach implies that, for every model for which an OLS fitting algorithm is available, the present approach yields a WLS fitting algorithm. In the special case where the WLS weight matrix is binary, the approach reduces to missing data imputation.

Key words: weighted least squares, alternating least squares, missing data, algorithms, majorization, matrix approximation, maximum likelihood estimation.

Many methods for multivariate analysis involve fitting a model to a data matrix. Often, the model is fit in the ordinary least squares (OLS) sense. That is, let $\mathbf{X}$ denote a given $n \times m$ data matrix and let $\mathbf{M}$ denote an $n \times m$ model description for these data. Then least squares fitting of this model to the data consists of minimizing

$$f(\mathbf{M}|\mathbf{X}) = \|\mathbf{X} - \mathbf{M}\|^2 \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - m_{ij})^2 \tag{1}$$

over $\mathbf{M}$, subject to certain constraints on $\mathbf{M}$. For instance, in case of principal components analysis (PCA), $\mathbf{M}$ is constrained to have rank r, where r is a fixed value, usually smaller than $\min(m, n)$. One way of dealing with this constraint is by writing $\mathbf{M} = \mathbf{AB}'$ for an $n \times r$ matrix $\mathbf{A}$ and an $m \times r$ matrix $\mathbf{B}$, and reformulating the problem as that of minimizing

$$g(\mathbf{A}, \mathbf{B}|\mathbf{X}) = \|\mathbf{X} - \mathbf{AB}'\|^2. \tag{2}$$

In various applications, it is desired to fit a model by minimizing a weighted least squares (WLS) loss function. For instance, Bailey and Gower (1990, also see ten Berge & Kiers, 1993) propose to approximate a symmetric matrix by a matrix of low rank while using differential weighting for the diagonal elements. Gabriel and Zamir (1979) approximate an asymmetric matrix by a matrix of low rank (as in (2)), while using differential weighting for all elements. Similarly, differential weights for the elements are used by Carroll, De Soete and Pruzanksy's (1989) procedure for WLS fitting of an N-way array by a multilinear model, and by R. A. Harshman's (Personal Communication, October 14, 1994) procedure for fitting the trilinear model to a three-way array (which, in fact, generalizes one of Gabriel and Zamir's procedures). A different type of application of WLS is in Verboon's (1994; see also Verboon & Heiser, 1992, 1994) series of methods for robust multivariate analysis: He proposes to minimize certain robust loss functions by iteratively minimizing a WLS loss function.

Besides the above explicit applications of WLS, some methods exist that are implicitly

This research has been made possible by a fellowship from the Royal Netherlands Academy of Arts and Sciences to the author. Requests for reprints should be sent to Henk A. L. Kiers, Department of Psychology (SPA), Grote Kruisstraat 2/1, 9712 TS Groningen, THE NETHERLANDS.

0033-3123/97/0600-95264$00.75/0

based on WLS. For instance, a common approach for fitting data which contain missing values (e.g., Commandeur, 1991; Gifi, 1990; ten Berge, Kiers & Commandeur, 1993) consists of setting the missing data elements initially at arbitrary values, and, in the loss function, assigning zero weights to the residuals that belong to these elements. Such loss functions can be minimized by a "missing data imputation" approach, which is a special instance of the EM algorithm (Dempster, Laird, & Rubin, 1977): By alternately fitting the model to the full data set (including estimates for the missing values), and replacing the missing elements by the current model estimates for these elements, the weighted loss function is decreased monotonically (and assumed to be minimized at least locally). Another implicit application of WLS fitting is maximum likelihood estimation in cases where the residuals are assumed to be independently normally distributed with zero mean and prespecified variance (see Verboon, 1994, pp. 38–40). In such cases maximum likelihood estimation is equivalent to WLS fitting with the weights taken equal to the inverse of the standard deviations. Also, WLS is sometimes used for maximum likelihood fitting in cases where the variance is not prespecified. In such cases the weights are adjusted in each iterative cycle. This is done, for instance, in the Carroll et al. (1989) procedure.

In all these methods, a WLS loss function is minimized. Let the weights (which are considered fixed and nonnegative here) be collected in an $n \times m$ matrix $\mathbf{W}$. Then the WLS loss function can be described generally as

$$\mathrm{h}(\mathbf{M}|\mathbf{X}, \mathbf{W}) = \|(\mathbf{X} - \mathbf{M})*\mathbf{W}\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^2(x_{ij} - m_{ij})^2, \tag{3}$$

where * denotes the Hadamard (or elementwise) product. Here, as in (1), $\mathbf{M}$ represents a large variety of models that are obtained by various choices of constraints on $\mathbf{M}$. The problem of minimizing the WLS function (3) is often much more complicated than that of minimizing the OLS function (1). Although special algorithms are available for some of the above mentioned WLS problems, there is a multitude of models for which WLS algorithms have not (yet) been proposed. Rather than resorting to general gradient based optimization techniques, which usually depend heavily on the availability of a good starting configuration, in the present paper an approach is used that is based on OLS. Specifically, a general procedure is offered that can be used to obtain algorithms for WLS fitting of *every* model for which an OLS fitting algorithm is available. This procedure decreases $\mathrm{h}(\mathbf{M}|\mathbf{X}, \mathbf{W})$ monotonically, until a (possibly local) minimum is obtained. This is achieved by iteratively decreasing $\mathrm{f}(\mathbf{M}|\tilde{\mathbf{X}})$, where $\tilde{\mathbf{X}}$ is a matrix that depends on $\mathbf{X}$, $\mathbf{W}$ and the values of $\mathbf{M}$ at the current iteration. It will be shown how $\tilde{\mathbf{X}}$ must be taken in order to ascertain that by decreasing $\mathrm{f}(\mathbf{M}|\tilde{\mathbf{X}})$ the function $\mathrm{h}(\mathbf{M}|\mathbf{X}, \mathbf{W})$ is decreased as well. In this way, the complicated WLS problem is solved by iteratively solving a, usually much simpler, OLS problem.

The main feature of the present approach for WLS fitting is that it allows one to fit every model by WLS for which an OLS fitting algorithm is available. The general idea of this approach, to be called iterative OLS here, has recently been described by Heiser (1995; see section 8.4, which also contains references to earlier work); in his terminology, iterative OLS solves a least distance problem in a nonidentity metric by repeatedly solving an unweighted problem by standard methods (p. 177). It is not claimed that the iterative OLS approach is the best one for every WLS fitting problem. Special purpose WLS fitting algorithms (for certain special models) are likely to be more efficient than the present general algorithm. Similarly, it is conceivable that the general method and the special purpose algorithms differ in sensitivity to local optima. To give some insight in these matters, the iterative OLS algorithm will be compared to some special purpose WLS algorithms. Specifically, we will compare the iterative OLS approach to Gabriel and Zamir's (1979) criss-cross multiple regression algorithm for WLS fitting of the PCA model, and to Verboon's approach for WLS orthogonal Procrustes rotation. Also, we sill study the value of the iterative OLS method for WLS fitting of the DEDICOM model a case where no special purpose algorithm seems to be available. We will start, however, by describing the iterative OLS procedure.

WLS Fitting by Means of Iterative OLS

The main purpose of the present paper is to investigate a general approach for WLS fitting (that is, minimizing (3)), in cases where an algorithm for OLS fitting (minimizing or decreasing (1)) is available. We will study an algorithm, which, starting from certain initial values for $\mathbf{M}$, updates the values in $\mathbf{M}$ iteratively. Suppose, at iteration i, the values in $\mathbf{M}$ are given by $\mathbf{M}_i$. Then, given $\mathbf{M}_i$, a function $k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W})$ will be defined that majorizes $h(\mathbf{M}|\mathbf{X}, \mathbf{W})$ and "touches it" at $\mathbf{M} = \mathbf{M}_i$ (i.e., $k(\mathbf{M}_i|\mathbf{M}_i, \mathbf{X}, \mathbf{W}) = h(\mathbf{M}_i|\mathbf{X}, \mathbf{W})$). The majorizing function, which is simpler than the original one, is minimized (or at least decreased) over $\mathbf{M}$ to obtain an update for $\mathbf{M}$. The crucial result that we will use is that, by minimizing such a majorizing function, we decrease the objective function, as has been shown, for instance, by de Leeuw and Heiser (1980) and Heiser (1987). This majorization principle underlies the general WLS algorithm investigated here.

To find the majorizing function $k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W})$, the objective function $h(\mathbf{M}|\mathbf{X}, \mathbf{W})$ is written as a special case of the function for which Heiser (1987, p. 344) derived a majorizing function. Upon definition of $\mathbf{D}_W$ as the diagonal matrix with on the diagonal the elements of Vec($\mathbf{W}$), where Vec($\mathbf{W}$) denotes the vector with the columns of $\mathbf{W}$ below each other, the function $h(\mathbf{M}|\mathbf{X}, \mathbf{W})$ can be rewritten as

$$h(\mathbf{M}|\mathbf{X}, \mathbf{W}) = \|(\mathbf{X} - \mathbf{M})*\mathbf{W}\|^2 = \|\mathbf{D}_W(\mathrm{Vec}(\mathbf{X}) - \mathrm{Vec}(\mathbf{M}))\|^2. \quad (4)$$

Heiser (1987, p. 345, Eq. (19)) derived a function that majorizes and touches a function of $\mathbf{Z}$ of the form $\mathrm{tr}(\mathbf{Y} - \mathbf{Z})'\mathbf{M}(\mathbf{Y} - \mathbf{Z})$; the fact that in his case $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ is irrelevant for his result. Clearly, (4) is of the same form as Heiser's function, hence we can use his majorization function here. In our case, this majorizing function is

$$k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W}) = \alpha + \beta\|\mathrm{Vec}(\mathbf{M}_i) + \beta^{-1}\mathbf{D}_W^2(\mathrm{Vec}(\mathbf{X}) - \mathrm{Vec}(\mathbf{M}_i)) - \mathrm{Vec}(\mathbf{M})\|^2, \quad (5)$$

where α is a constant, and β is the largest eigenvalue of $\mathbf{D}_W^2$, which is the maximum of the squared elements of $\mathbf{W}$, denoted as w_m^2. Substituting w_m^2 for β in (5), and writing Vec($\cdot$)'s as matrices again, we obtain

$$k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W}) = \alpha + w_m^2\|\mathbf{M}_i + w_m^{-2}\mathbf{W}^{(2)}*\mathbf{X} - w_m^{-2}\mathbf{W}^{(2)}*\mathbf{M}_i - \mathbf{M}\|^2 = \alpha + w_m^2 f(\mathbf{M}|\tilde{\mathbf{X}}_i), \quad (6)$$

where $\tilde{\mathbf{X}}_i \equiv (\mathbf{M}_i + w_m^{-2}\mathbf{W}^{(2)}*\mathbf{X} - w_m^{-2}\mathbf{W}^{(2)}*\mathbf{M}_i)$, and $\mathbf{W}^{(2)}$ denotes $\mathbf{W}*\mathbf{W}$. Thus, we have obtained a function that majorizes and touches the WLS function $h(\mathbf{M}|\mathbf{X}, \mathbf{W})$, and that itself is related in a simple way to $f(\mathbf{M}|\tilde{\mathbf{X}}_i)$, the loss function corresponding to OLS fitting of the same model $\mathbf{M}$ to a matrix $\tilde{\mathbf{X}}_i$ rather than $\mathbf{X}$.

When a (closed-form) solution is available for the OLS problem, that is, for minimizing, or at least decreasing, $f(\mathbf{M}|\tilde{\mathbf{X}}_i)$ and hence $k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W})$, we can construct an algorithm for monotonically decreasing $h(\mathbf{M}|\mathbf{X}, \mathbf{W})$, as follows. Let $\mathbf{M}_i$ denote the current values for $\mathbf{M}$, and let $\mathbf{M}_{i+1}$ denote the values for $\mathbf{M}$ that minimize (or decrease) $f(\mathbf{M}|\tilde{\mathbf{X}}_i)$ and hence $k(\mathbf{M}|\mathbf{M}_i, \mathbf{X}, \mathbf{W})$. Then we have

$$h(\mathbf{M}_{i+1}|\mathbf{X}, \mathbf{W}) \leq k(\mathbf{M}_{i+1}|\mathbf{M}_i, \mathbf{X}, \mathbf{W}) < k(\mathbf{M}_i|M_i, \mathbf{X}, \mathbf{W}) = h(\mathbf{M}_i|\mathbf{X}, \mathbf{W}), \quad (7)$$

hence, by updating $\mathbf{M}$ in this way, the function h is decreased. By iteratively minimizing or decreasing $f(\mathbf{M}|\tilde{\mathbf{X}}_i)$, we have a monotonically decreasing sequence of function values, which must converge because the function value is bounded below by zero. Thus, we have a monotonically convergent algorithm for WLS fitting. Hence, as soon as we have a procedure for OLS fitting of a particular model, we can construct an algorithm for WLS fitting of the same model by iterative application of the OLS procedure. It should be noted that this procedure cannot be guaranteed to lead to a *global* minimum of the WLS loss function. To avoid the chance of taking a locally optimal solution for a globally optimal one, it is recommended to run the algorithm from several different starts and assume that the best of the resulting solutions gives the global optimum.

Schematically, we can summarize the iterative OLS procedure as follows:

Step 1. Initialize $\mathbf{M}_i$ as $\mathbf{M}_0$; set $i = 0$ and compute $h_0 = h(\mathbf{M}_0|\mathbf{X}, \mathbf{W})$.

Step 2. Compute $\tilde{\mathbf{X}}_i = (\mathbf{M}_i + w_m^{-2}\mathbf{W}^{(2)}*\mathbf{X} - w_m^{-2}\mathbf{W}^{(2)}*\mathbf{M}_i)$.

Step 3. Compute $\mathbf{M}_{i+1}$ as the $\mathbf{M}$ that decreases or minimizes $\|\tilde{\mathbf{X}}_i - \mathbf{M}\|^2$ subject to the constraints on $\mathbf{M}$.

Step 4. Compute $h_{i+1} = h(\mathbf{M}_{i+1}|\mathbf{X}, \mathbf{W})$; if $h_i - h_{i+1} > \varepsilon * h_i$ (for a prespecified small value ε) set $i = i + 1$ and go to Step 2; else consider the algorithm converged.

A particularly simple choice for initializing $\mathbf{M}$ is by setting $\mathbf{M}_0 = \mathbf{0}$, despite the fact that this choice will often not satisfy the constraints on $\mathbf{M}$. After one iteration, the constraints on $\mathbf{M}$ will be satisfied, and $\mathbf{M}_1$ will be the OLS solution for fitting $\mathbf{M}$ to $w_m^{-2}\mathbf{W}^{(2)}*\mathbf{X}$, hence this procedure implicitly amounts to starting the iterative OLS algorithm by the OLS solution for a weighted version of the data. An alternative possibility is to initialize $\mathbf{M}$ by the OLS solution for the unweighted data.

Missing Data Imputation

A common application of WLS is in the context of handling missing data, where unit weights are used for nonmissing values, and zero weights for missing values. It will now be shown that, when such binary weights are employed, the algorithm described above reduces to a common missing data imputation algorithm. In other words, the above algorithm is a direct generalization of the missing data imputation procedure.

In a missing data imputation algorithm, we start by certain estimates for the missing data, fit the model to the complete data (including the estimates for missing data), adjust the missing data by equating them to their model estimates, fit the model to the updated data, etcetera. By defining $\mathbf{W}$ as the binary matrix with zeros indicating missing values, this procedure can be described more formally as:

Step 1′. Initialize $\mathbf{M}_i$ as $\mathbf{M}_0$; set $i = 0$ and compute $h_0 = h(\mathbf{M}_0|\mathbf{X}, \mathbf{W})$. (*The function value is computed as the sum of squared residuals for the non-missing values.*)

Step 2′. Compute $\tilde{\mathbf{X}}_i = \mathbf{W}*\mathbf{X} + \mathbf{W}^c*\mathbf{M}_i$, where $\mathbf{W}^c$ contains the binary complements of $\mathbf{W}$. (*Missing values are set at their current model estimates.*)

Step 3′. Compute $\mathbf{M}_{i+1}$ as the $\mathbf{M}$ that decreases or minimizes $\|\tilde{\mathbf{X}}_i - \mathbf{M}\|^2$ subject to the constraints on $\mathbf{M}$. (*The model is fitted to the updated data matrix.*)

Step 4′. Compute $h_{i+1} = h(\mathbf{M}_{i+1}|\mathbf{X}, \mathbf{W})$; if $h_i - h_{i+1} > \varepsilon * h_i$ set $i = i + 1$ and go to Step 2; else consider the algorithm converged.

Comparing this algorithm to the iterative OLS algorithm described above, the only difference is between Steps 2 and 2′. However, when the iterative OLS procedure is used with a binary matrix $\mathbf{W}$, then $w_m^2 = 1$ and Step 2 reduces to "Compute $\tilde{\mathbf{X}}_i = (\mathbf{M}_i + \mathbf{W}*\mathbf{X} - \mathbf{W}*\mathbf{M}_i)$", which is readily verified to be equivalent to Step 2′. Thus, it follows that, in the special case where $\mathbf{W}$ is binary, the iterative OLS procedure reduces to a missing data imputation procedure.

The iterative OLS algorithm has now been seen to incorporate missing data imputation as a special case. However, it is known that, at least in certain special cases (e.g., see ten Berge & Kiers, 1989), missing data imputation is not a very efficient procedure. In the remainder of this paper, we will consider some aspects of the performance of iterative OLS in a variety of model fitting problems.

WLS fitting of the PCA model

The first problem to be considered is that of fitting the bilinear model $\mathbf{M} = \mathbf{AB}'$ (as in (2)) to a data matrix, with weights taken arbitrarily. This problem was described by Gabriel and Zamir (1979), and they proposed to handle it by (iterative) criss-cross algorithms. We used the criss-cross multiple regression approach, which they found to perform best. The idea of this approach is that first the matrix **A** is updated by means of rowwise weighted least squares regressions, next, matrix **B** is updated by rowwise weighted least squares regressions, then again **A** is updated, etc. This clearly differs from the iterative OLS approach studied here, since that approach updates **A** and **B** simultaneously, via PCA of the matrix $\tilde{\mathbf{X}}$.

Both algorithms for this special case were programmed in PCMATLAB and applied to 30 (randomly constructed) data matrices **X** (ten of size 100 × 20, ten of size 500 × 20, and ten of size 100 × 20), using 30 different, randomly constructed weight matrices **W** of corresponding sizes. The random elements of **X** were drawn from the uniform [−.2,.8] distribution, those of **W** from the uniform [0, 1] distribution. The former choice was made to ensure that the expected PCA solutions are neither dominated by a size factor (as they would when all data values were nonnegative) nor reflecting a complete absence of relations between columns of **X** (as they would when data values were taken from a symmetric distribution). The choice for the weights had no special motivation, except that it was ensured that weights are nonnegative. For the analysis of the first twenty data sets (of sizes 100 × 20 and 500 × 20), we took the dimensionality equal to 3. For the other ten (100 × 20) data sets, dimensionality 6 was taken. In each analysis we used the OLS PCA solution as a start. The convergence parameter ε was set to 10^{-6}.

For each analysis, the final loss function value, as well as the total number of floating point operations (flops; as defined in PCMATLAB) were recorded. In 28 out of 30 cases the resulting function values differed less than 1%, and were therefore considered equal. In the two other cases, the iterative OLS procedure led to the lowest function value, and apparently, the criss-cross regression algorithm led to a local minimum.

To see to what extent these differences are caused by the fact that only one start is used, we reanalyzed each data set by both methods, using a random start for **A** and **B**, and for each method inspected the best out of the two function values. It turned out that the criss-cross algorithm in five cases yielded a function value that was more than 1% lower than in the original analyses, and the iterative OLS algorithm did so in four cases. The net result is that, in 29 cases the two algorithms yielded virtually equal function values, and in only one case the iterative OLS algorithm led to a function value that was more than 1% lower than that resulting from the criss-cross algorithm. It can be concluded that the methods hardly differ in terms of sensitivity to local minima, and it can be expected that this difference will further diminish when more than one additional randomly started run is used.

As far as computational efficiency is concerned, the methods do differ considerably: The iterative OLS algorithm required 3 to 4 times more flops than the criss-cross algorithm did. This is in line with our experience that alternating least squares algorithms usually converge faster than majorization based algorithms. It can be concluded that there is little reason to replace the criss-cross regression by our procedure. It can likewise be expected that Harshman's generalization of criss-cross regression to WLS fitting of the trilinear model is better than iterative OLS.

The present conclusion of the superiority of criss-cross regression may suggest that iterative OLS does not have much practical value for cases where WLS fitting algorithms already exist. However, in the next example, it will be seen that our algorithm is considerably more efficient than the existing one.

Orthogonal Procrustes Rotation by WLS

The orthogonal Procrustes rotation problem (Green, 1952) can be seen as fitting a model $\mathbf{M} = \mathbf{XT}$, with $\mathbf{X}$ fixed and $\mathbf{T}$ constrained to be orthonormal, to a particular (target) matrix $\mathbf{Y}$. In the context of robustifying Procrustes rotation, Verboon (1994, pp. 52–55) described two procedures for Procrustes rotation by WLS. The first procedure handles the situation where all rows get the same weight (see Verboon & Heiser, 1992), which has a straightforward solution. The second procedure handles WLS with elementwise differential weights. This case can be described as minimizing

$$\sigma(\mathbf{T}|\mathbf{X}, \mathbf{Y}, \mathbf{W}) = \|(\mathbf{Y} - \mathbf{XT})*\mathbf{W}\|^2 \tag{8}$$

over $\mathbf{T}$, subject to $\mathbf{T}'\mathbf{T} = \mathbf{I}$, where $\mathbf{X}$ and $\mathbf{Y}$ are given $n \times m$ matrices and $\mathbf{W}$ is an $n \times m$ matrix of weights (which are, in fact, the square roots of Verboon's (nonnegative) weights). It should be noted that, in Verboon's context, the weights depend on the residuals ($\mathbf{Y} - \mathbf{XT}$), but this is ignored in the present context. Verboon has reformulated the function as

$$\sigma(\mathbf{T}|\mathbf{X}, \mathbf{Y}, \mathbf{W}) = \sum_{i=1}^{n} \|(\mathbf{y}_i' - \mathbf{x}_i'\mathbf{T})\mathbf{W}_i\|^2, \tag{9}$$

where $\mathbf{x}_i'$ and $\mathbf{y}_i'$ denote the horizontal (row) vectors corresponding to the i-th rows of $\mathbf{X}$ and $\mathbf{Y}$, respectively, and $\mathbf{W}_i$ denotes the diagonal matrix with the elements of row i of $\mathbf{W}$ on its diagonal. Verboon's algorithm for minimizing (9) is a special case of Kiers' (1990) general majorization algorithm, as follows. By expanding (9) as

$$\sigma(\mathbf{T}|\mathbf{X}, \mathbf{Y}, \mathbf{W}) = c - 2\sum_{i=1}^{n} \operatorname{tr} \mathbf{W}_i^2\mathbf{y}_i\mathbf{x}_i'\mathbf{T} + \sum_{i=1}^{n} \operatorname{tr} \mathbf{x}_i\mathbf{x}_i'\mathbf{T}\mathbf{W}_i^2\mathbf{T}', \tag{10}$$

it is written in the same way as Kiers' general function. According to Kiers (1990, p. 421), to decrease (10), $\mathbf{T}$ can be updated as $\mathbf{PQ}'$, where $\mathbf{P}$ and $\mathbf{Q}$ are taken from the singular value decomposition (SVD)

$$\mathbb{T} - \left(2\sum_{i=1}^{n} \alpha_i\right)^{-1}\left(-2\sum_{i=1}^{n} \mathbf{x}_i\mathbf{y}_i'\mathbf{W}_i^2 + 2\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i'\mathbb{T}\mathbf{W}_i^2\right) = \mathbf{PDQ}', \tag{11}$$

where $\mathbb{T}$ denotes the current $\mathbf{T}$, and α_i is taken (larger than or) equal to the largest eigenvalue of $\mathbf{x}_i\mathbf{x}_i' \otimes \mathbf{W}_i^2$; the matrices $\mathbf{P}$, $\mathbf{D}$ and $\mathbf{Q}$ defined by the SVD are orthonormal, diagonal, and orthonormal, respectively. It follows that $\gamma \equiv \Sigma_i\alpha_i$ can be taken as $\gamma \equiv \Sigma_i\mu_i\mathbf{x}_i'\mathbf{x}_i$, where μ_i denotes the maximum squared element of row i of $\mathbf{W}$. Expression (11) can be simplified as

$$\mathbb{T} + \gamma^{-1}\left(\sum_{i=1}^{n} \mathbf{x}_i\mathbf{y}_i'\mathbf{W}_i^2 - \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i'\mathbb{T}\mathbf{W}_i^2\right) = \mathbf{PDQ}'. \tag{12}$$

Upon noting that $\mathbf{y}_i'\mathbf{W}_i^2$ and $\mathbf{x}_i'\mathbb{T}\mathbf{W}_i^2$ contain the rows of $\mathbf{W}^{(2)}*\mathbf{Y}$ and $(\mathbf{W}^{(2)}*\mathbf{X}\mathbb{T})$, respectively, we can rewrite (12) as

$$\mathbb{T} + \gamma^{-1}\mathbf{X}'(\mathbf{W}^{(2)}*\mathbf{Y}) - \gamma^{-1}\mathbf{X}'(\mathbf{W}^{(2)}*\mathbf{X}\mathbb{T}) = \mathbf{PDQ}'. \tag{13}$$

Hence, in Verboon's algorithm the update for $\mathbf{T}$ is given by $\mathbf{PQ}'$ from the SVD in (13).

The iterative OLS procedure studied in the present paper gives an alternative algorithm for WLS Procrustes rotation. Specifically, using the iterative OLS approach, an update for $\mathbf{T}$ is found by minimizing

$$\|\mathbf{X}\mathbb{T} - w_m^{-2}(\mathbf{W}^{(2)}*(\mathbf{X}\mathbb{T}) + w_m^{-2}\mathbf{W}^{(2)}*\mathbf{Y} - \mathbf{XT}\|^2, \tag{14}$$

where w_m is the maximum of the squared elements of $\mathbf{W}$. The minimum of (14) over $\mathbf{T}$ is obtained by taking $\mathbf{T} = \tilde{\mathbf{P}}\tilde{\mathbf{Q}}'$, where $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$ are taken from the SVD

$$\mathbf{X}'\mathbf{X}\mathbb{T} - w_m^{-2}\mathbf{X}'(\mathbf{W}^{(2)}*\mathbf{X}\mathbb{T}) + w_m^{-2}\mathbf{X}'(\mathbf{W}^{(2)}*\mathbf{Y}) = \tilde{\mathbf{P}}\tilde{\mathbf{D}}\tilde{\mathbf{Q}}', \tag{15}$$

see Cliff (1966).

Verboon's algorithm and the iterative OLS algorithm (based on (13) and (15), respectively) differ in only two respects: First, where the update in (13) has $\mathbb{T}$, the one in (15) has $\mathbf{X}'\mathbf{X}\mathbb{T}$, and second, the γ in (13) is replaced by w_m^2 in (15). This difference is similar to that between Kiers' (1990) majorization algorithm and Kiers and ten Berge's (1992) refined majorization algorithm. For that case it could be proven that the refined majorization function is closer to the object function than Kiers' original majorization function, and it was seen that the refined majorization approach tends to converge more quickly. Here, Verboon's algorithm is essentially based on Kiers' majorization function, and the iterative OLS algorithm is based on a function differing only slightly from Kiers and ten Berge's majorizing function. For that reason, we may expect the iterative OLS algorithm to converge more quickly than Verboon's algorithm.

In a small simulation study, the performance of the two algorithms was compared. Forty pairs of matrices $\mathbf{X}$ and $\mathbf{Y}$ were constructed (of sizes 20×4, 20×8, 40×4 and 40×8). The elements of the matrices $\mathbf{X}$ were randomly drawn from the uniform $[-.2,.8]$ distribution. Rather than taking $\mathbf{Y}$ completely at random, it was chosen to construct $\mathbf{Y}$ from $\mathbf{X}$ by rotating $\mathbf{X}$ by a random rotation matrix, and adding a random matrix (from the uniform $[-.2,.8]$ distribution). For every pair $(\mathbf{X}, \mathbf{Y})$, a random weights matrix $\mathbf{W}$ was chosen (from the uniform $[0, 1]$ distribution). For each of the 40 triples $(\mathbf{X}, \mathbf{Y}, \mathbf{W})$, a WLS Procrustes analysis was done, using Verboon's algorithm and the iterative OLS algorithm. For each analysis, we used a rational start for $\mathbf{T}$ (based on the OLS Procrustes solution), and ten random starts; in all cases the convergence criterion ε was set to 10^{-6}. The best of the eleven runs was considered "the solution" of the method at hand. The best of the two solutions (that is, obtained by Verboon's method and by the iterative OLS algorithm) was considered the "globally optimal" solution, even though it is conceivable that even in as many as 22 runs the global optimum is still not found.

It turned out that Verboon's algorithm was (on the average) more than ten times as slow as the iterative OLS approach, and that in all cases the WLS solution had a lower (and hence better) function value. Hence, indeed the algorithm resembling refined majorization turned out to be more efficient than Verboon's one (based on ordinary majorization). The differences in function value were small in all cases except one. In the latter case Verboon's solution probably was a local minimum, whereas in the other cases, the differences are probably due only to the fact that Verboon's algorithm takes smaller steps: In the vicinity of the optimum Verboon's algorithm stops earlier than the iterative OLS algorithm, because the convergence criterion is satisfied earlier. The only advantage of Verboon's algorithm over the iterative OLS algorithm was that its individual runs led to local optima less frequently: With Verboon's algorithm, the rational start led to a local optimum only once, and random starts led to local optima only 38 (out of 400) times, whereas with the iterative OLS algorithm, the rational start led to a local optimum 4 times (out of 40) and the random starts did so 75 (out of 400) times. However, as mentioned above, the best solution out of eleven runs with Verboon's algorithm was never better than that obtained by the iterative OLS algorithm, and in fact, in one case the eleven runs of Verboon's algorithm missed the global optimum. On the basis of our simulation, it can be concluded that the iterative OLS approach is far more efficient than Verboon's method, and, provided that a multistart procedure is used, at least as reliable in avoiding local optima.

As has been mentioned, Verboon's algorithm is based on ordinary majorization, and the iterative OLS algorithm is similar to refined majorization. The difference between these algorithms can also be described as follows: Verboon's algorithm is based on majorizing a function of $\mathbf{T}$, whereas the general WLS algorithm is based on majorizing a function of $\mathbf{M} = \mathbf{XT}$. The former approach requires, as we have seen, elaborating the function at hand as a function of $\mathbf{T}$, whereas the latter can be adopted without such elaboration. In fact, all fitting problems discussed by Verboon (1994) could have been solved by using the majorization of a function of $\mathbf{M}$ (rather than of the unknown parameters determining $\mathbf{M}$), where $\mathbf{M}$ denotes the model at hand, because the iterative OLS approach can be used for *any* technique for which an OLS solution is available.

WLS fitting of the DEDICOM Model

Above, we have discussed WLS fitting problems for which algorithms have already been proposed in the literature. We will now discuss fitting a model for which this is not the case. The DEDICOM model (Harshman, 1978) represents an asymmetric but square data matrix $\mathbf{X}$ ($n \times n$) as

$$\mathbf{X} = \mathbf{ARA}', \tag{16}$$

where $\mathbf{A}$ is an $n \times r$ "loading" matrix of the n objects on r main aspects in the data, and $\mathbf{R}$ is an $r \times r$ matrix that represents the (a)symmetric relations between these main aspects. For a description of the model we refer to Harshman, Green, Wind and Lundy (1982). A first monotonically convergent algorithm for OLS fitting of this model has been given by Kiers (1989) and a more efficient one by Kiers, ten Berge, Takane and de Leeuw (1990). For general WLS fitting of the DEDICOM model, no monotonically convergent algorithm has as yet been proposed, but it is obvious that iterative OLS can be used here. In fact, the special case where $\mathbf{W} = \mathbf{11}' - \mathbf{I}$ (leading to off-diagonal fitting of the DEDICOM model) has already been dealt with by Takane (1985) and ten Berge and Kiers (1989).

In the present section, we will evaluate in a small simulation study the use of WLS for fitting the DEDICOM model. Specifically, we have constructed a 7×7 data set to which the DEDICOM model fits perfectly in three dimensions, and to each data element we added random noise from a centered uniform distribution with a range proportional to the size of the data value. This choice of noise generation was meant to simulate the situation where the measurement error increases with the size of the data values, as can be expected, for instance, in the car switching data reported by Harshman et al. (1982), where the data range from values like 0, 4, 6, to high values like 61350, 63509, 67964 and 81808. In such cases it seems reasonable that, if the DEDICOM model fits well, the misfit should be much smaller for the small values than for the high values. Using OLS will not necessarily lead to such solutions. Therefore, in such cases, it seems reasonable to fit the DEDICOM model in terms of WLS, with weights taken equal to, for instance, the inverse of the data elements. This approach has been tested on the above artificial data set, constructed such that the ranges of the distributions of the model errors are indeed proportional to the size of the data values. In this way, we are able to check the validity of the solutions of OLS and WLS DEDICOM.

The artificial data **X** were constructed on the basis of the following matrices $\mathbf{A}_c$, $\mathbf{R}_c$ and the noise-free data matrix $\mathbf{X}_c = \mathbf{A}_c\mathbf{R}_c\mathbf{A}_c'$:

$$\mathbf{A}_c = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 3 \\ 0 & 0 & 3 \end{pmatrix} \mathbf{R}_c = \begin{pmatrix} 1 & 0 & 4 \\ 5 & 4 & 6 \\ 4 & 1 & 4 \end{pmatrix} \mathbf{X}_c = \begin{pmatrix} 1 & 1 & 0 & 0 & 12 & 12 & 12 \\ 1 & 1 & 0 & 0 & 12 & 12 & 12 \\ 10 & 10 & 16 & 16 & 36 & 36 & 36 \\ 10 & 10 & 16 & 16 & 36 & 36 & 36 \\ 12 & 12 & 6 & 6 & 36 & 36 & 36 \\ 12 & 12 & 6 & 6 & 36 & 36 & 36 \\ 12 & 12 & 6 & 6 & 36 & 36 & 36 \end{pmatrix}.$$

A random matrix **N** was constructed (with elements drawn from a uniform $[-.1,.1]$ distribution), and each element of **N** was multiplied by the corresponding element of $\mathbf{X}_c$, before it was added to $\mathbf{X}_c$. Thus, the data matrix **X** was obtained as $\mathbf{X} = \mathbf{X}_c + \mathbf{N}*\mathbf{X}_c$:

$$\mathbf{X} = \begin{pmatrix} 0.90 & 1.04 & 0 & 0 & 11.80 & 11.59 & 12.53 \\ 0.93 & 1.04 & 0 & 0 & 12.48 & 12.32 & 12.61 \\ 10.51 & 10.87 & 16.55 & 17.38 & 38.95 & 37.85 & 37.09 \\ 9.92 & 9.77 & 14.42 & 17.11 & 37.89 & 39.54 & 32.92 \\ 12.08 & 12.05 & 5.86 & 6.03 & 34.29 & 35.03 & 36.95 \\ 11.33 & 12.79 & 5.48 & 5.51 & 32.74 & 34.18 & 38.77 \\ 10.91 & 10.88 & 5.90 & 6.18 & 37.70 & 39.47 & 34.36 \end{pmatrix}.$$

The above data set was analyzed by DEDICOM, using three dimensions, as well as by WLS DEDICOM. In the latter case, the weight matrix was defined by $w_{ij} = x_{ij}^{-1}$ if $x_{ij} \neq 0$, and $w_{ij} = 1$ if $x_{ij} = 0$. Note that this choice establishes that the residuals for the smallest elements are taken most seriously. The DEDICOM results were obtained by the Kiers et al. (1990) algorithm, using a rational start as well as two random starts, and a convergence criterion of .0001% of the function value. The three runs gave the same function value, and all used only few iterations (e.g., the run started from the rational start required only 13 iterations). The solution yielded the data estimates given in Table 1. In the second panel of Table 1, the residuals ($\mathbf{X} - \mathbf{ARA}'$) of the OLS fitting of the DEDICOM model are given. It can be seen that the residuals range from .04 to 2.55 (in absolute sense), and that small and large residuals are spread more or less evenly over the data: The absolute residuals for small data elements are not systematically smaller than those for large data elements as expressed by their correlation with the data values of $-.05$. Clearly, the OLS solution does not take into account the differences in accuracy by which the data elements satisfy the three-dimensional DEDICOM model.

Although the residuals do not correspond to the structure in the true residuals, it is still possible that the solution reflects the elements in the original matrices $\mathbf{A}_c$ and $\mathbf{R}_c$ fairly well. Before the solution for **A** and **R** is considered, we remark that the obtained **A** is determined up to a nonsingular transformation only (which can be compensated by recomputing **R**). The common way to exploit this freedom is to enhance interpretability by varimax rotation of the (columnwise orthonormal) matrix **A**. The resulting **A** is reported in Table 2. It can be seen that this solution for **A** does not correspond very well to the matrix $\mathbf{A}_c$ from which the data have been constructed. However, it is still conceivable that a transformation of **A** can be found that better corresponds to $\mathbf{A}_c$. The one that corresponds best can be obtained via regression of the columns of **A** on those of $\mathbf{A}_c$. The resulting transformed solution is given in Table 2 as the "**A**, after matching **A** to $\mathbf{A}_c$". The associated matrix **R** is given as well. It can be seen that, even after this transformation to optimal

TABLE 1

(OLS) DEDICOM Estimates and Residuals for Artificial Data

Data Estimates (ARA′)						
2.32	2.23	0.81	0.88	12.04	12.55	9.98
2.24	2.11	0.75	0.79	12.70	13.27	10.21
10.03	10.18	16.85	16.80	38.32	39.12	36.23
9.59	9.70	15.92	15.86	37.57	38.40	35.13
12.12	12.74	5.69	6.76	34.13	34.96	37.06
12.25	12.91	5.26	6.40	33.59	34.39	36.97
10.23	10.42	4.63	5.31	37.65	38.91	35.92
Residuals (X–ARA′)						
−1.42	−1.19	−0.81	−0.88	−0.24	−0.96	2.55
−1.32	−1.08	−0.75	−0.79	−0.22	−0.95	2.40
0.48	0.69	−0.30	0.58	0.63	−1.28	0.86
0.33	0.07	−1.49	1.25	0.32	1.13	−2.21
−0.04	−0.69	0.17	−0.73	0.16	0.07	−0.11
−0.92	−0.12	0.22	−0.89	−0.84	−0.21	1.80
0.68	0.47	1.27	0.87	0.05	0.57	−1.56

agreement with $\mathbf{A}_c$, the DEDICOM solutions for $\mathbf{A}$ and $\mathbf{R}$ differ considerably from the matrices $\mathbf{A}_c$ and $\mathbf{R}_c$, respectively, on the basis of which the data were constructed: For instance, the first two rows of $\mathbf{A}$, as well as the seventh, deviate considerably from those of $\mathbf{A}_c$.

Having seen that the OLS DEDICOM solution does not reproduce the original $\mathbf{A}$ and $\mathbf{R}$ very well, we now turn to the WLS DEDICOM solution. To minimize the WLS DEDICOM function, we can iteratively apply one step of the OLS DEDICOM algorithm to the matrix $\bar{\mathbf{X}}_i = (\mathbf{M}_i + w_m^{-2}\mathbf{W}^{(2)}*\mathbf{X} - w_m^{-2}\mathbf{W}^{(2)}*\mathbf{M}_i)$, where $\mathbf{M}_i = \mathbf{A}_i\mathbf{R}_i\mathbf{A}_i'$ and $\mathbf{A}_i$ and $\mathbf{R}_i$ are the i-th updates of $\mathbf{A}$ and $\mathbf{R}$, respectively. In one such step the WLS DEDICOM function is decreased by updating both the matrices $\mathbf{A}$ and $\mathbf{R}$. However, it is not difficult to improve the update for $\mathbf{R}$, by using WLS regression. Therefore, we updated $\mathbf{R}$ by means of WLS regression. The thus constructed algorithm was started by the same rational start for $\mathbf{A}$ and $\mathbf{R}$ as in the OLS DEDICOM analysis, and the same convergence criterion was used. The algorithm converged after 4459 iterations. Although this number may seem prohibitive, the fact that each iteration is very efficient implies that such a process takes still only 5 minutes on a machine with 486, 66MHz processor (and would take considerably less when programmed in a more efficient language than in the interpreter based language PCMATLAB). In addition to this rationally started run, we ran the algorithm 10 times from random starting positions. In all cases we found the same loss function value (.103).

TABLE 2

(OLS) DEDICOM Solution for A and R

A, After Varimax Rotation of A		
−0.13	−0.01	0.45
−0.17	−0.01	0.52
0.02	0.73	−0.04
−0.02	0.69	0.05
0.66	0.01	0.05
0.71	−0.01	0.02
0.14	0.01	0.73
A, After Matching A to A_c		
0.47	−0.02	0.48
0.55	−0.02	0.44
−0.06	2.06	0.01
0.04	1.94	0.02
−0.15	0.03	3.09
−0.20	−0.05	3.24
0.66	0.03	2.38
R, After Matching A to A_c		
−2.34	0.04	4.60
4.12	4.08	6.31
4.36	0.97	3.94

The estimates from the rationally started run, as well as the residuals, are presented in Table 3. It can be seen that now, indeed, the residuals for the small elements are much smaller than those for the large elements, as is also expressed by the fact that the absolute residuals correlated .75 with the data values. It is more important, however, to see to what extent the underlying structure is recovered. Table 4 gives the solution for **A**, first after varimax and next after transforming it to optimal agreement with $\mathbf{A}_c$. For the latter solution, the corresponding matrix **R** is given as well. It can be seen that already after varimax rotation (thus without using the usually unavailable information on the underlying matrix $\mathbf{A}_c$), the obtained **A** is very similar to $\mathbf{A}_c$. In fact, the columns have congruence coefficients of .9997, .9995, and .9944, respectively, with the corresponding columns of $\mathbf{A}_c$. The solution for **A** and **R** after matching **A** to $\mathbf{A}_c$ is very similar indeed to the underlying structure given by $\mathbf{A}_c$ and $\mathbf{R}_c$, and much better than the OLS solution.

The present (artificial) example demonstrates two things. First, it illustrates that data

TABLE 3

WLS DEDICOM Estimates and Residuals for Artificial Data

Data Estimates (**ARA**′)						
0.90	0.99	0.01	0.01	12.21	12.23	11.74
0.95	1.05	-0.01	-0.01	12.47	12.49	11.99
10.32	10.65	16.73	16.43	37.51	37.92	37.80
9.85	10.16	15.93	15.64	36.38	36.76	36.62
11.61	11.97	5.86	6.16	34.88	35.71	35.40
11.36	11.73	5.39	5.69	34.96	35.78	35.42
11.31	11.69	5.79	6.07	35.98	36.78	36.39
Residuals (**X**–**ARA**)						
0.00	0.04	-0.01	-0.01	-0.41	-0.64	0.79
-0.02	-0.01	0.01	0.01	0.01	-0.17	0.61
0.19	0.22	-0.18	0.95	1.44	-0.07	-0.71
0.07	-0.40	-1.50	1.47	1.51	2.77	-3.70
0.47	0.07	-0.00	-0.12	-0.59	-0.68	1.54
-0.03	1.07	0.09	-0.18	-2.21	-1.60	3.35
-0.40	-0.80	0.11	0.12	1.72	2.70	-2.02

where noise is proportional to the data values can be fit much better by WLS than by OLS. Second, the iterative OLS approach to WLS requires very many iterations. This is probably due to the fact that the weights in **W** differ considerably (by more than a factor 40). This phenomenon was encountered even more severely when we used the iterative OLS method to fit Harshman et al.'s (1982) car switching data in three dimensions. For this analysis we only used 14 of the 16 car segments (dropping the very small segments SMAC and LUXI). When using weights equal to the inverses of the data values, the algorithm converged (with $\varepsilon = 10^{-6}$) only after 38690 iterations; this run was started by using the best solution resulting from six runs with $\varepsilon = 10^{-4}$. The normalized varimax rotated solution, reported in Table 5 (with labels as used by Harshman et al.), can be seen to differ substantially from the normalized varimax rotated OLS solution (which is very similar to the solution of DEDICOM on the full data matrix). Rather than giving a substantive interpretation of the results, we will focus on the differences between the solutions. The main difference is between the second dimensions. It should be noted that this difference cannot be removed by regressing one solution on the other. Specifically, it can be seen that in the OLS solution, the second dimension is dominated by the car segments STDM and LUXD, which are quite large car segments, with relatively many switchings within their own segment. This idiosyncracy, reflecting mainly the higher marginal values for STDM and LUXD, has disappeared in the WLS solution. The second WLS dimension focuses on

TABLE 4

WLS DEDICOM Solution for **A** and **R**

A after Varimax Rotation		
0.70	0.00	−0.00
0.71	−0.00	0.00
−0.02	0.72	0.00
0.02	0.69	−0.00
−0.07	0.00	0.59
0.00	−0.02	0.59
0.07	0.01	0.56
A, After Matching **A** to $\mathbf{A}_c$		
0.98	0.00	−0.00
0.99	−0.00	0.02
−0.03	2.05	0.01
0.03	1.95	−0.01
−0.10	0.01	3.06
0.00	−0.04	3.04
0.10	0.04	2.90
R, After Matching **A** to $\mathbf{A}_c$		
0.93	−0.01	4.08
5.13	4.05	6.20
3.88	0.97	3.97

other aspects of the data (related mainly to the car segments SUBD, SUBI, SMAI, COMI and MIDI). The main conclusion here is that, by using WLS rather than OLS, we obtain a solution providing substantively different information, which is, moreover, no longer predominantly determined by large marginal values.

For both solutions we also computed the correlation between the data and the associated (absolute) residuals. For the OLS solution we found a value of .72 (which is relatively high, since no such relation is enforced or stimulated by the OLS procedure); for the WLS solution the correlation is as high as .87. It can be concluded that, if for the car switching data the size of the residuals should be strongly associated with the size of the data values (which seems a reasonable assumption for the present frequency data), the WLS solution should be preferred. Furthermore, the WLS solution is preferable because it does not allow objects with large marginal frequencies to dominate part of the solution.

TABLE 5

OLS and WLS DEDICOM Solution for Car Switching Data

	A after Normalized Varimax Rotation					
	OLS			WLS		
SUBD	.14	−.05	**.35**	.08	**−.34**	**.61**
SUBC	.02	−.01	.03	.03	.02	.03
SUBI	.03	−.02	**.30**	−.08	**.58**	**.33**
SMAD	.02	−.04	**.52**	.01	.09	**.48**
SMAI	.00	.00	.09	−.04	**.44**	.01
COML	**.25**	−.11	.15	**.26**	−.07	.13
COMM	.11	.00	.05	.18	−.03	.03
COMI	.02	.00	.03	−.01	**.31**	−.02
MIDD	**.54**	.04	.12	**.64**	.22	.01
MIDI	.02	.00	.02	−.01	**.30**	−.02
MIDS	.08	.17	**.61**	−.01	.00	**.50**
STDL	**.68**	.01	−.20	**.56**	−.17	−.05
STDM	**.25**	**.73**	−.19	**.39**	.18	−.11
LUXD	**−.29**	**.65**	.14	.06	.19	−.02

	Associated **R** (divided by 1000)					
	OLS			WLS		
dim. 1	120	50	81	68	15	83
dim. 2	28	95	33	11	9	16
dim. 3	18	15	79	19	10	54

Discussion

With the availability of a general algorithm for WLS fitting of models for which OLS fitting procedures are already available, a host of new WLS fitting possibilities has emerged. These possibilities range from the simple orthogonal Procrustes model, to more complex models as the DEDICOM model, and even to considerably more complicated models as the DYNAMALS model by Bijleveld and de Leeuw (1991), or the PARAFAC2 model (see Harshman & Lundy, 1984; Kiers, 1993). For the latter two models no WLS fitting algorithms seem to have been proposed, but OLS algorithms are available. By using the iterative OLS approach, it has become possible to fit those models in the WLS sense. Furthermore, if one is willing to assume independent normal error distributions with known variances, one can obtain ML estimates for these models as well. In fact, even without the assumption of known variances, one can use the present approach in a procedure for Iteratively Reweighted Least Squares to *approximate* ML solutions (e.g., as in Carroll et al., 1989). In such cases, the weights are updated after each cycle of updating the model parameters.

In analogy to the present approach to solving WLS problems, it is possible to minimize Generalized Least Squares (GLS) functions (e.g., see Bollen, 1989), which are often used to fit linear structural relation models (LISREL), by iterative OLS minimization. In some preliminary test analyses, however, we did not see a clear advantage in using this approach over the standard procedures in programs as LISREL8 (Jöreskog & Sörbom, 1993). For instance, GLS fitting of the common factor model might be performed by iteratively using (one step of) the MINRES (Harman & Jones, 1966) algorithm, but for this fitting problem, the LISREL program worked fine. However, an alternative GLS fitting model is potentially useful for situations where the LISREL program does not work well.

An important other type of GLS functions can be handled similarly: The class of GLS functions that yield asymptotically distribution free estimates, proposed by Browne (1984). By applying the same principle to this type of GLS function, Browne's GLS problem can be reduced to iterative OLS problems in almost the same way as we did in the previous section. In this way, considerably larger problems can be handled than with Browne's procedure, since the iterative use of an $m(m + 1)/2 \times m(m + 1)/2$ matrix, where m denotes the total number of variables, can then be avoided.

In some situations, a fitting problem cannot be described in terms of fitting a model to fixed data. In particular, in some cases the data have to be "fitted" as well, for instance, in the context of optimal scaling (see Gifi, 1990). In such cases, our approach can still be used: By alternately fitting the model to the data and the data to the model, the WLS function can be minimized by fitting the associated OLS problems (to associated matrices $\tilde{\mathbf{X}}$).

The iterative OLS method studied in this paper is particularly attractive because of its omnipotence: It can handle *all* problems for which OLS procedures are available. However, this is not to say that the algorithm *should* be used universally for solving WLS problems. For instance, for fitting the linear model, a closed-form WLS solution is available, and there is no use in replacing this solution by one obtained via an iterative process, which is not even guaranteed to give the global minimum. Also, as we have seen, iterative OLS can be terribly slow. In such cases, more efficient algorithms should be used, if they are available.

References

Bailey, R. A., & Gower, J. C. (1990). Approximating a symmetric matrix. *Psychometrika*, *55*, 665–675.

Bijleveld, C. & de Leeuw, J. (1991). Fitting longitudinal reduced rank regression models by alternating least squares. *Psychometrika*, *56*, 443–447.

Bollen, K. A. (1989). *Structural equations with latent variables.* New York: Wiley.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology*, *37*, 62–83.

Carroll, J. D., De Soete, G., & Pruzansky, S. (1989). Fitting of the latent class model via iteratively reweighted least squares candecomp with nonnegativity constraints. In R. Coppi & S. Bolasco (Eds.), *Multiway data analysis* (pp. 463–472). Amsterdam: Elsevier Science Publishers.

Cliff, N. (1966). Orthogonal rotation to congruence. *Psychometrika*, *31*, 33–42.

Commandeur, J. J. F. (1991). Matching configurations. Leiden: DSWO Press.

de Leeuw, J. & Heiser, W. (1980). Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah (Ed.), *Multivariate analysis V* (pp. 501–522). Amsterdam: North Holland.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, *Series B*, *39*, 1–38.

Gabriel, K. R., & Zamir, S. (1979). Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, *21*, 489–498.

Gifi, A. (1990). *Nonlinear multivariate analysis*. Chichester: Wiley.
Green, B. F. (1952). The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika, 17*, 429–440.
Harman, H. H., & Jones, W. H. (1966). Factor analysis by minimizing residuals (Minres). *Psychometrika, 31*, 351–368.
Harshman, R. A. (1978, August). *Models for analysis of asymmetrical relationships among N objects or stimuli.* Paper presented at the First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology, Hamilton, Ontario.
Harshman, R. A., Green, P. E., Wind, Y., & Lundy, M. E. (1982). A model for the analysis of asymmetric data in marketing research, *Marketing Science, 1*, 205–242.
Harshman, R. A., & Lundy, M. E. (1984). The PARAFAC model for three-way factor analysis and multidimensional scaling. In H. G. Law, C. W. Snyder, J. A. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp. 122–215). New York: Praeger.
Heiser, W. J. (1987). Correspondence Analysis with least absolute residuals. *Computational Statistics and Data Analysis, 5*, 337–356.
Heiser, W. J. (1995). Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. In W. J. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis* (pp. 157–189). Oxford: Oxford University Press.
Jöreskog, K. G., & Sörbom, D. (1993). *LISREL 8 User's guide*. Chicago: Scientific Software International.
Kiers, H. A. L. (1989). An alternating least squares algorithm for fitting the two- and three-way DEDICOM model and the IDIOSCAL model. *Psychometrika, 54*, 515–521.
Kiers, H. A. L. (1990). Majorization as a tool for optimizing a class of matrix functions. *Psychometrika, 55*, 417–428.
Kiers, H. A. L. (1993). An alternating least squares algorithm for PARAFAC2 and DEDICOM3. *Computational Statistics and Data Analysis, 16*, 103–118.
Kiers, H. A. L., & ten Berge, J. M. F. (1992). Minimization of a class of matrix trace functions by means of refined majorization. *Psychometrika, 57*, 371–382.
Kiers, H. A. L., ten Berge, J. M. F., Takane, Y., & de Leeuw, J. (1990). A generalization of Takane's algorithm for DEDICOM. *Psychometrika, 55*, 151–158.
Takane, Y. (1985). Diagonal Estimation in DEDICOM. *Proceedings of the 1985 Annual Meeting of the Behaviormetric Society* (pp. 100–101). Sapporo, Japan: Behaviormetric Society.
ten Berge, J. M. F., & Kiers, H. A. L. (1989). Fitting the off-diagonal DEDICOM model in the least-squares sense by a generalization of the Harman & Jones MINRES procedure of factor analysis. *Psychometrika, 54*, 333–337.
ten Berge, J. M. F., & Kiers, H. A. L. (1993). An alternating least squares method for the weighted approximation of a symmetric matrix. *Psychometrika, 58*, 115–118.
ten Berge, J. M. F., Kiers, H. A. L., & Commandeur, J. J. F. (1993). Orthogonal Procrustes rotation for matrices with missing values. *British Journal of Mathematical and Statistical Psychology, 46*, 119–134.
Verboon, P. (1994). *A robust approach to nonlinear multivariate analysis*. Leiden: DSWO Press.
Verboon, P., & Heiser, W. J. (1992). Resistant orthogonal Procrustes analysis. *Journal of Classification, 9*, 237–256.
Verboon, P., & Heiser, W. J. (1994). Resistant lower rank approximation of matrices by iterative majorization. *Computational Statistics and Data Analysis, 18*, 457–467.

Manuscript received 1/26/95
Final version received 1/19/96